UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/691,450	10/23/2003	Scott Hanggie	306212.01/MFCP.140739	6405

45809 7590 11/18/2009

SHOOK, HARDY & BACON L.L.P.
(c/o MICROSOFT CORPORATION)
INTELLECTUAL PROPERTY DEPARTMENT
2555 GRAND BOULEVARD
KANSAS CITY, MO 64108-2613

EXAMINER	
AMIN, JWALANT B	

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
11/18/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/691,450	HANGGIE ET AL.	
	Examiner	Art Unit	
	JWALANT AMIN	2628	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 June 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,7-11,15-24,27-31 and 37-48* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4,7-11,15-24,27-31 and 37-48* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *06/29/2009*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. It should be noted that the applicant has not responded to claims 15-16 rejected under 35 U.S.C. 112, second paragraph, in the reply filed on 6/29/2009. The applicant's reply should fully address all rejections, objections or any other concerns raised in the previous office action. Though the reply filed on 6/29/2009 is considered, failure to address all issues, in future reply, will result in a non-responsive amendment. The rejection for claims 15-16 under 35 U.S.C. 112, second paragraph, is maintained. Please refer below for details.

2. Applicant's arguments filed 6/29/2009 have been fully considered but they are not persuasive.

3. Regarding claims 1, 19, 21 and 39, the applicant argues that Morgenstern, Portuesi, Erickson, Apple Computers, Lipton, Lyons, and Moki, fail to describe or suggest "... receiving, at a desktop window manager (DWM), application content from legacy applications in a top-to-bottom order to display the application content received in a top- to-bottom order in windows corresponding to the legacy application in the graphical user interface; stripping out application content received from the legacy applications; converting the stripped application content to a graphical representation; switching between the CDWM and the DWM to render the advanced application content and legacy application content" (see pgs. 13 and 15-17 of applicant's remarks). The applicant further argues "...Nothing in Morgenstern describes or suggests interfacing with legacy application content." (see pg. 14 and pg. 17). The applicant further argues

"...Portuesi fails to describe or suggest legacy application that is received in the top-to-bottom order is rendered in the top-to-bottom order in a legacy application window." (see pg. 14 and pg. 17). The applicant further argues "... Apple Computers fails to describe or suggest rendering windows associated with legacy applications and advanced applications in a graphical user interface." (see pg. 14 and pg. 17). The applicant further argues "...Lipton does not describe receiving legacy application content and rendering that content. ... Lipton fails to describe or suggest receiving the legacy application content in top-to-bottom order." (see pg. 14 and pg. 17). The applicant further argues "...Lyons fails to describe rendering windows associated with legacy applications and advanced applications in a graphical user interface." (see pg. 14). The applicant further argues "...nothing in Moki describes a rendering engine capable of interface with both legacy applications and advanced applications simultaneously" (see pg. 14).

4. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

It should be further noted that merely stating that individual references do not teach or suggest the limitations as claimed, amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

In this instant case, the examiner interprets that Morgenstern teaches a computer implemented method for rendering a desktop window in a graphical user interface of an operating system shell, comprising: receiving application content, at a compositing desktop window manager (CDWM) (Quartz Compositor) (receiving the application content, pg. 1 last paragraph, pg. 3 third and fourth paragraphs), corresponding to the advanced applications in the graphical user interface (Quartz compositor takes information from the rendering component and writes it on the screen, Morgenstern: pg. 1 last paragraph; it should be noted that Morgenstern further teaches QuickDraw handles text, vector graphics and bitmapped images, and then send them to the screen and output devices, Morgenstern: pg. 1 fourth paragraph). Morgenstern further teaches the windows having translucent frame portions (translucent title bars of inactive windows, Morgenstern: pg. 2 third paragraph).

However, Morgenstern does not explicitly teach receiving application content from advance applications in a bottom-to-top order (video from VL is passed on to OpenGL by converting it into bottom-to-top orientation from top-to-bottom orientation, Portuesi: pg. 2-3), to display the application content received in a bottom-to-top order in windows (openGL renders in bottom-to-top orientation, Portuesi: pg. 2; Quartz 2D renders drawing primitives, PDF documents, text and images using bottom-to-top operation, Lindberg: pg. 1). However, Portuesi teaches exactly the same.

Morgenstern and Portuesi do not explicitly teach receiving, at a desktop window manager (DWM) (QuickDraw), application content information from legacy applications (Carbon/Cocoa application) (picture window is a Carbon application which uses

QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D, Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; it should be noted that examiner takes an official notice of the fact that QuickDraw is a legacy API from classic Mac OS). However, Erickson teaches exactly the same. Erickson further teaches stripping out application content from the legacy window content (QuickDraw receives the picture content from the picture window of the Carbon application, Erickson: pg. 1 last two lines and pg. 2 first three lines), and converting the application content to a graphical representation of the application content (QuickDraw generates the graphical representation of the application data using it's drawing methods; picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D, Erickson: pg. 1 last two lines and pg. 2 first four paragraphs). It should be further noted that Erickson teaches switching between the CDWM and the DWM to render the advanced application content and legacy application content (picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D; Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; it should be noted that switching application content between DWM and CDWM is functionally equivalent to redirecting application content from DWM to CDWM).

Morgenstern, Portuesi and Erickson do not explicitly teach receiving application content in a top-to-bottom order (application content in the form of a print job specifies the layout direction left to right then top to bottom, Apple2: pg. 14; it should be noted that left to right then top to bottom is functionally equivalent to top to bottom order; print

job consists of drawing commands and printing system can receive drawing commands from an application in several ways including Carbon applications using QuickDraw, Apple2: pg. 24-25). However, Apple2 teaches exactly the same.

Morgenstern, Portuesi, Erickson and Apple2 do not explicitly teach displaying the application content in a top-to-bottom order (QuickDraw renders in top-to-bottom order, Lipton: pg. 4 figure 3 and pg. 5 first three lines) in windows corresponding to the legacy application in the graphics user interface. However, Lipton teaches exactly the same.

Morgenstern, Portuesi, Erickson, Apple2 and Lipton do not explicitly teach switching between the CDWM and the DWM to render the advanced application content and legacy application (Cocoa application that uses Quartz 2D does not provide all the needed functionality, so there is a switching between Quartz and QuickDraw for some things, Lyons: pg. 1). However, Lyons teaches exactly the same.

Morgenstern, Portuesi, Erickson, Apple2, Lipton and Lyons do not explicitly teach displaying at least a portion of the application content in an opaque content portion of the windows (fully opaque window with transparent sub-parts, Moki: sixth paragraph). However, Moki teaches exactly the same. Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to use different functionalities of different versions of Jaguar operating system as taught by Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki because combining different functionalities will result in a better and more user-friendly operating system.

5. Regarding claims 1, 19, 21 and 39, the applicant further argues that Erickson fails to teach "… stripping out application content from the received legacy window content" (see pg. 15 and pg. 17).

6. However, the examiner disagrees. The examiner interprets that Erickson teaches stripping out (receiving) application content (picture content) from the legacy window content (picture window of Carbon application) (QuickDraw receives the picture content from the picture window of the Carbon application, Erickson: pg. 1 last two lines and pg. 2 first three lines). See rejection of claim 1 for further details.

7. Regarding claims 1 and 19, the applicant argues "…In Lyons, at pg. 1, it appears Cocca uses the AppKit not Quartz 2d. Moreover, in Apple Computers, at pg. 25, it appears Cocca uses Coca drawing routines, which are separate from Quartz 2d. These inconsistencies suggest that the references do not refer to the same version of the operating system. Accordingly, it is improper to use these references to support the obviousness rejection because they do not refer to the same release or product." (see pg. 15).

8. However, the examiner disagrees. The examiner interprets that none of the references of Lyons and Apple2 (Apple Computers) teach that the alternative routines (AppKit and Quartz 2d) could not be on the same version of an operating system. It should be further noted that none of the reference criticize, discredit or otherwise discourage the solution claimed. The examiner interprets that an operating system that had both the routines (ApptKit or Quartz 2d) available at its disposal to be used by

Cocoa, could choose to use either the AppKit or the Quartz 2d depending on the requirement.

9. Regarding claims 21 and 39, the applicant further argues "... nothing in Portuesi, Erickson, Apple Computers, Lipton, Lyons, and Moki, describes or suggests, among other things, redirecting application content from the DWM to the CDWM to render content in a graphical user interface" (see pg. 17).

10. However, the examiner disagrees. The examiner interprets that Erickson teaches switching between the CDWM and the DWM to render the advanced application content and legacy application content (picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D; Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; it should be noted that switching application content between DWM and CDWM is functionally equivalent to redirecting application content from DWM to CDWM). Please refer below to the rejection of claim 21 for details.

11. Regarding claim 44, the applicant argues Erickson, in view of Lyons and further in view of Siracusa, fails to describe or suggest "... providing legacy window information from an instance of a legacy application program to a legacy desktop window manager (DWM) executing on the computer; stripping out client content from the legacy window information; converting the client content to a raster image of the client content; ... applying a texture to a mesh, and wherein the texture comprises the raster image of the client content and default non-client information" (see pg. 20). The applicant also argues "... the references relied on by the Office do not provide a coherent and consistent

description of Jaguar". The applicant further argues "... the references fail to describe converting legacy application content to a raster image that is used to texture a mesh created by a compositing window manager. (see pg. 20). The applicant further argues that Erickson, Lyons and Siracusa fail to describe "... rendering legacy application content as the texture of a mesh" and "...the default non-client information that is included in the texture" (see pg. 20-21). The applicant further argues that Erickson, Lyons and Siracusa do not suggest "... the interaction between a legacy DWM and CDWM to render content in a graphical user interface" (see pg. 21). The applicant further argues that Erickson fails to teach "... stripping out application content from the received legacy window content" (see pg. 20).

12. However, the examiner disagrees. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

It should be further noted that merely stating that individual references do not teach or suggest the limitations as claimed, amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

In this instant case, the examiner interprets that Erickson, Lyons and Siracusa describe different versions of Jaguar operating system that teach the limitations as stated in claim 44. It would have been obvious to one of ordinary skill in the art to use

different functionalities of different versions of Jaguar operating system as taught by Erickson, Lyons and Siracusa because combining different functionalities will result in a better and more user-friendly operating system.

In this instant case, the examiner interprets that Erickson teaches the instance of the legacy application program (Carbon/Cocoa application) providing legacy window information from an instance of a legacy application program to a legacy desktop window manager (DWM) (QuickDraw) executing on the computer (picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D, Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; it should be noted that examiner takes an official notice of the fact that QuickDraw is a legacy API from classic Mac OS). Erickson further teaches stripping out client content from the legacy window information (QuickDraw receives the picture content from the picture window of the Carbon application, Erickson: pg. 1 last two lines and pg. 2 first three lines), and converting the client content to raster image of the client content (QuickDraw generates the graphical representation of the application data using it's drawing methods; picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D, Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; it should be also noted that picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D, Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; It should be noted that Cocoa application that uses

Quartz 2D does not provide all the needed functionality, so there is a switching between Quartz and QuickDraw for some things, Lyons: pg. 1).

Although Erickson teaches the limitations as stated, Erickson does not explicitly teach a compositing desktop window manager (CDWM) (Quartz 2D and Quartz compositor), executing on the computer, drawing a window to a buffer memory (Siracusa: pg. 1 last paragraph, pg. 2 first two paragraphs and figure on pg. 3), wherein the CDWM renders the window by applying a texture to a mesh (window/polygon) (it should be noted that a mesh according to the specification is 2D or 3D primitive, see paragraph [0015] on pg. 15; it should be further noted that each window is treated as an OpenGL surface and the texture is mapped onto that surface, Siracusa: pg. 3 second paragraph and pg. 4 first five lines). However Siracusa teaches exactly the same (the window server, now an OpenGL application itself, retains the resulting bitmaps as textures on polygons in an OpenGL scene and composites them into a pleasing, cohesive final image on the screen, Siracusa: pg. 3 second paragraph). Siracusa further teaches wherein the texture comprises the raster image (bitmap) of the client content and the default non-client information (bitmap includes translucency and anti-aliasing information, Siracusa: pg. 1 last paragraph, pg. 2 first two paragraphs, pg. 3 second paragraph, pg. 4 first paragraph; all of the bitmapped data produced by QuickDraw is passed on to the Quartz Compositor for eventual display on the screen, Siracusa: pg. 1 seventh paragraph, figure on pg. 3 and 4). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to combine the teachings of Siracusa into the operating system of Erickson and Lyons because such an operating

system will composite the resulting bitmaps as textures on polygons and composite them into a pleasing, cohesive final image on the screen.

13. Regarding claim 41, the applicant argues that Morgenstern and Moki in view of Solazzi, Whitman and Fowler, fail to suggest "... receiving application content to display in a window; ... the compositing desktop window manager is configured to provide transparency, shadows, lighting effects, bump mapping, and environment mapping" (see pg. 23). The applicant further argues "... nothing in Morgenstern describes or suggests environmental windowing effects" (see pg. 23). The applicant argues "... nothing in Moki describes or suggests a rendering engine capable of providing environmental mappings" (see pg. 23). The applicant also argues that Fowler does not teach "... a windows manager that provides bump mappings and environmental mappings" (see pg. 23).

14. However, the examiner disagrees. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

It should be further noted that merely stating that individual references do not teach or suggest the limitations as claimed, amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

In this instant case, the examiner interprets that Morgenstern teaches receiving application content to display in a window (Quartz compositor takes information from the rendering component and writes it on the screen, Morgenstern: pg. 1 last paragraph). Morgenstern further teaches window having a frame portion (translucent title bars of inactive windows, Morgenstern: pg. 2 third paragraph). Morgenstern further teaches the compositing desktop window manager is configured to provide transparency and shadows (Quartz's window server makes it easy to see the outlines and shadings of buttons and other window elements through the translucent title bars of inactive windows, Morgenstern: pg. 2 paragraph 3, figure 1).

Although Morgenstern teaches the limitations as stated, Morgenstern does not explicitly teach displaying at least a portion of the application content in a content portion of the window having a frame portion (fully opaque window with transparent sub-parts, Moki: sixth paragraph; transparent sub-parts such as title bars, shadows, etc., Moki: sixth paragraph; Moki further teaches fully opaque windows has transparent sub-parts such as title bar, the shadows, etc, Moki: paragraphs 3, 4, 6 and 9). . Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to use different functionalities of different versions of Jaguar operating system as taught by Morgenstern and Moki because combining different functionalities will result in a better and more user-friendly operating system.

Although Morgenstern and Moki teach the limitations as stated, they do not explicitly teach rendering spectral highlights on the frame portion based on a virtual light source by the compositing desktop window manager configured to provide light effects.

However, Solazzi teaches a 3D image (window frame) can display reflective and refractive characteristics ([0008]; it should be noted that the unit performing 3D modeling corresponds to compositing desktop window manager; it should be noted that Whitman defines spectral highlight as a bright reflection from a light source containing little or no detail; Solazzi teaches a 3D image can display reflective characteristics, and therefore the reflective characterisitics of the 3D image of Solazzi also includes spectral highlights which is a bright reflection form a light source; spectral highlights correspond to lighting effects; therefore, compositing desktop window manager provides lighting effects). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have 3D images with refractive characteristics as taught by Solazzi and apply into the operating system of Morgenstern and Moki because refractive properties added to the 3D image makes the image appear more realistic and reflective characteristics illustrate the ability of the object to reflect light ([0008], [0014]).

Although Morgenstern, Moki, Solazzi and Whitman teach the limitations as stated above, they do not explicitly the compositing desktop window manager is configured to provide bump mapping and environment mapping. However, Fowler teaches OpenGL supports environment-mapped bump mapping (environment-mapped bump mapping corresponds to bump mapping and environment mapping, [0038-0039]). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to provide bump mapping and environment mapping as taught by Fowler to the method Morgenstern, Moki, Solazzi and Whitman because such an approach will

achieve a rendered surface whose appearance changes more realistically with lighting conditions ([0038]).

15. Regarding claim 43, the applicant argues that Morgenstern, Portuesi and Moki, in view of Solazzi, in view of Whitman, in view of Fowler and further in view of Ben-Shachar fails to suggest "... receiving, at a compositing desktop window manager, application content in reverse z-order to display in a window; ... the compositing desktop window manager is configured to provide transparency, shadows, lighting effects, bump mapping, and environmental mapping" (see pg. 25). The applicant further argues "... Morgenstern, Portuesi and Moki describes some window transparency but fails to describe or suggest environmental mapping" (see pg. 25). The applicant further argues that Fowler fails to teach "... a windows manager that provides bump mappings and environmental mappings" and that Ben-Shachar fails to teach "... a compositing desktop window manager that receives application content in reverse z-order" (see pg. 25).

16. However, the examiner disagrees. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

It should be further noted that merely stating that individual references do not teach or suggest the limitations as claimed, amount to a general allegation that the

claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

In this instant case, the examiner interprets that Morgenstern teaches receiving application content, at a compositing desktop window manager (CDWM) (receiving the application content is taught by Morgenstern: pg. 1 last paragraph, pg. 3 third and fourth paragraphs). It should be noted that Morgenstern further teaches QuickDraw handles text, vector graphics and bitmapped images, and then send them to the screen and output devices (Quartz compositor takes information from the rendering component and writes it on the screen, Morgenstern: pg. 1 fourth paragraph and last paragraph). Morgenstern further teaches the windows having translucent frame portions (translucent title bars of inactive windows, Morgenstern: pg. 2 third paragraph). Morgenstern further teaches the compositing desktop window manager is configured to provide transparency and shadows (Quartz's window server makes it easy to see the outlines and shadings of buttons and other window elements through the translucent title bars of inactive windows, Morgenstern: pg. 2 paragraph 3, figure 1).

However, Morgenstern does not explicitly teach receiving application content in a bottom-to-top order (video from VL is passed on to OpenGL by converting it into bottom-to-top orientation from top-to-bottom orientation, Portuesi: pg. 2-3), to display the application content received in a bottom-to-top order in windows (openGL renders in bottom-to-top orientation, Portuesi: pg. 2; Quartz 2D renders drawing primitives, PDF documents, text and images using bottom-to-top operation, Lindberg: pg. 1). However, Portuesi teaches exactly the same.

Although Morgenstern and Portuesi teach the limitations as stated, they do not explicitly teach displaying at least a portion of the application content in a content portion of the window having a frame portion (fully opaque window with transparent sub-parts, Moki: sixth paragraph; transparent sub-parts such as title bars, shadows, etc., Moki: sixth paragraph; Moki further teaches fully opaque windows has transparent sub-parts such as title bar, the shadows, etc, Moki: paragraphs 3, 4, 6 and 9). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to use different functionalities of different versions of Jaguar operating system as taught by Morgenstern, Portuesi and Moki because combining different functionalities will result in a better and more user-friendly operating system.

Although Morgenstern, Portuesi and Moki teaches the limitations as stated above, they do not explicitly teach rendering refractive content on the frame portion based on the other discrete content behind the window in the graphical user interface by the compositing desktop window manager, which is configured to provide light effects. However, Solazzi teaches a 3D image (window frame) can display reflective and refractive characteristics ([0008]; it should be noted that the unit performing 3D modeling corresponds to compositing desktop window manager; it should be noted that Whitman defines spectral highlight as a bright reflection from a light source containing little or no detail; Solazzi teaches a 3D image can display reflective characteristics, and therefore the reflective characterisitics of the 3D image of Solazzi also includes spectral highlights which is a bright reflection form a light source; spectral highlights correspond to lighting effects; therefore, compositing desktop window manager provides lighting

effects). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have 3D images with refractive characteristics as taught by Solazzi and apply into the operating system of Morgenstern, Portuesi and Moki because refractive properties added to the 3D image makes the image appear more realistic and reflective characteristics illustrate the ability of the object to reflect light ([0008], [0014]).

Although Morgenstern, Portuesi, Moki, Solazzi and Whitman teach the limitations as stated above, they do not explicitly the compositing desktop window manager is configured to provide bump mapping and environment mapping. However, Fowler teaches OpenGL supports environment-mapped bump mapping (environment-mapped bump mapping corresponds to bump mapping and environment mapping, [0038-0039]). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to provide bump mapping and environment mapping as taught by Fowler to the method Morgenstern, Portuesi, Moki, Solazzi and Whitman because such an approach will achieve a rendered surface whose appearance changes more realistically with lighting conditions ([0038]).

Although Morgenstern, Portuesi, Moki, Solazzi, Whitman and Fowler teach the limitations as stated above, they do not explicitly teach receiving the application content in reverse z-order. However, Ben-Shachar teaches exactly the same ([0061]; it should be noted that DT_WINDLST packet containing a list corresponds to application content; it should be noted that the process proceeds through all windows in the windows list in reverse z-order). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to process through the windows list in reverse z-order as

taught by Ben-Shachar and apply it to the method of Morgenstern, Portuesi, Moki, Solazzi, Whitman and Fowler because treating the window information is such a manner will create the viewer display of shared window ([0061]).

17. Regarding claim 42, the applicant argues Morgenstern, Portuesi, Erickson, Apple Computer, Lipton, Lyons and Moki, in view of Solazzi, in view of Whitman, in view of Fowler and further in view of Ben-Shachar fails to suggest "... receiving, at a compositing desktop window manager, application content in reverse z-order to display in a window; ... the compositing desktop window manager is configured to provide transparency, shadows, lighting effects, bump mapping, and environmental mapping" (see pg. 26-27). The applicant further argues "... Morgenstern, Portuesi Erickson, Apple Computer, Lipton, Lyons and Moki fail to describe environmental mappings and receiving application content in reverse z-order" (see pg. 27). The applicant further argues that Fowler fails to teach "... a windows manager that provides bump mappings and environmental mappings" and that Ben-Shachar fails to teach "... a compositing desktop window manager that receives application content in reverse z-order" (see pg. 25).

18. However, the examiner disagrees. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

It should be further noted that merely stating that individual references do not teach or suggest the limitations as claimed, amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

The arguments presented for claim 42 are similar to that provided for claim 43, and therefore the examiner states the same reasons as presented above.

19. Regarding claim 45, the applicant argues that Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki, Siracusa, Farrah, and Meagher fails to describe "... dividing the mesh, associated with the window displayed by the computer, into three regions per mesh dimension; for each region, maintaining offsets of mesh vertices in any dimension by which the region is bounded by a bounding box of the window, and scaling mesh vertices in any dimension by which the region is not bounded by the bounding box of the window" (see pg. 30).

20. However, the examiner disagrees. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

It should be further noted that merely stating that individual references do not teach or suggest the limitations as claimed, amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

In this instant case, the examiner interprets that Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons, Moki and Siracusa teach the limitations as stated, except that they do not explicitly teach receiving user input to resize the window and dividing the mesh into the three regions per mesh dimension. However, Farrah teaches to resize the window by dividing it into several equally sized and not equally sized regions based on the user input (fig. 21a-c, fig. 22a-c, [0225-0227], [0231-0235]; it should be noted that selecting the number of rows as "3" will divide the window in nine equal-sized regions; it should be noted that dividing window is functionally equivalent to resizing the window; it should be noted that dividing window is functionally equivalent to resizing the window; it should be noted that the window is being divided along x and y dimensions). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to divide a window mesh into several regions as taught by Farrah and apply it into the method of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons, Moki and Siracusa because such regions are commonly used in computer programs which are used to generate artworks, drawings and flow charts ([0006]).

Although the combination of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons, Moki, Siracusa and Farrah teach the limitations as stated above, they do not explicitly teach for each region, maintaining offsets of mesh vertices in any dimension by which the region is bounded by a bounding box of the window, and scaling mesh vertices in any dimension by which the region is not bounded by the bounding box of the window. However, Meagher shows offsets (maintaining offsets of mesh vertices) from each line correspond to vertices of each of the four windows, and the critical

vertices for a window overlay selected from 3 x 3 array may be calculated by adding offsets (scaling the vertices) as a function of the value in x and y directions (fig. 6a-f, fig. 23a-e, col. 8 lines 31-51, col. 58 lines 64-67, col. 59 lines 1-15, col. 60 lines 3-21 and lines 56-67, col. 61 lines 1-17). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to scale the vertices as taught by Meagher and apply it into the method of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons, Moki, Siracusa and Farrah because such a method scales the three-dimensional universe relative to the three-dimensional coordinate system using the independent scaling factors for each of the x, y and z directions input by floating point multiplication on the microcomputer (col. 60 lines 56-60).

Claim Rejections - 35 USC § 112

21. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

22. Claims 15-16 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

23. Claim 15 recites the limitation "method of claim 13" in line 1. There is insufficient antecedent basis for this limitation in the claim. It should be noted that claim 13 has been cancelled.

24. Claim 16 recites the limitation "method of claim 13" in lines 1-2. There is insufficient antecedent basis for this limitation in the claim. It should be noted that claim 13 has been cancelled.

Claim Rejections - 35 USC § 103

25. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

26. Claims 1, 19, 21 and 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over David Morgenstern (Under the desktop: Prospecting for Quartz in Mac OS X; http://www.creativepro.com/story/feature/17439.html?origin=story; August 22, 2002; pgs. 1-4; hereinafter Morgenstern), Portuesi et al. (Displaying In-Memory Video Using OpenGL; http://www.lurkertech.com/lg/ogl.video.html; October 16, 2002; hereinafter Portuesi), Shawn Erickson (Screenshot PDF; http://www.omnigroup.com/mailman/archive/macosx-talk/2002-July/071171.html; July 30, 2002; hereinafter Erickson), Apple Computers (About the Mac OS X Printing System; December 11, 2002; hereinafter Apple2), Lipton (QuickDraw GX for Postscript programmers; http://www.mactech.com/articles/develop/issue_15/051-070_Lipton_final.html; August 19, 2000), Torrey Lyons (Re: MacOS X; http://www.xfree86.org/pipermail/forum/2003-July/003741.html; July 9, 2003; hereinafter Lyons), and further in view of Moki (Aqua help in Nvidia GeForce 4 [Archive] –

AppleInsider; http://forums.appleinsider.com/archive/index.php/t-1122.html; January 28, 2002; pg. 1),

27. Regarding claim 1, Morgenstern teaches a computer implemented method for rendering a desktop window in a graphical user interface of an operating system shell, comprising: receiving application content, at a compositing desktop window manager (CDWM) (Quartz Compositor) (receiving the application content, pg. 1 last paragraph, pg. 3 third and fourth paragraphs), corresponding to the advanced applications in the graphical user interface (Quartz compositor takes information from the rendering component and writes it on the screen, Morgenstern: pg. 1 last paragraph; it should be noted that Morgenstern further teaches QuickDraw handles text, vector graphics and bitmapped images, and then send them to the screen and output devices, Morgenstern: pg. 1 fourth paragraph). Morgenstern further teaches the windows having translucent frame portions (translucent title bars of inactive windows, Morgenstern: pg. 2 third paragraph).

However, Morgenstern does not explicitly teach receiving application content from advance applications in a bottom-to-top order (video from VL is passed on to OpenGL by converting it into bottom-to-top orientation from top-to-bottom orientation, Portuesi: pg. 2-3), to display the application content received in a bottom-to-top order in windows (openGL renders in bottom-to-top orientation, Portuesi: pg. 2; Quartz 2D renders drawing primitives, PDF documents, text and images using bottom-to-top operation, Lindberg: pg. 1). However, Portuesi teaches exactly the same.

Morgenstern and Portuesi do not explicitly teach receiving, at a desktop window manager (DWM) (QuickDraw), application content information from legacy applications (Carbon/Cocoa application) (picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D, Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; it should be noted that examiner takes an official notice of the fact that QuickDraw is a legacy API from classic Mac OS). However, Erickson teaches exactly the same. Erickson further teaches stripping out application content from the legacy window content (QuickDraw receives the picture content from the picture window of the Carbon application, Erickson: pg. 1 last two lines and pg. 2 first three lines), and converting the application content to a graphical representation of the application content (QuickDraw generates the graphical representation of the application data using it's drawing methods; picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D, Erickson: pg. 1 last two lines and pg. 2 first four paragraphs). It should be further noted that Erickson teaches switching between the CDWM and the DWM to render the advanced application content and legacy application content (picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D; Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; it should be noted that switching application content between DWM and CDWM is functionally equivalent to redirecting application content from DWM to CDWM).

Morgenstern, Portuesi and Erickson do not explicitly teach receiving application content in a top-to-bottom order (application content in the form of a print job specifies the layout direction left to right then top to bottom, Apple2: pg. 14; it should be noted that left to right then top to bottom is functionally equivalent to top to bottom order; print job consists of drawing commands and printing system can receive drawing commands from an application in several ways including Carbon applications using QuickDraw, Apple2: pg. 24-25). However, Apple2 teaches exactly the same.

Morgenstern, Portuesi, Erickson and Apple2 do not explicitly teach displaying the application content in a top-to-bottom order (QuickDraw renders in top-to-bottom order, Lipton: pg. 4 figure 3 and pg. 5 first three lines) in windows corresponding to the legacy application in the graphics user interface. However, Lipton teaches exactly the same.

Morgenstern, Portuesi, Erickson, Apple2 and Lipton do not explicitly teach switching between the CDWM and the DWM to render the advanced application content and legacy application (Cocoa application that uses Quartz 2D does not provide all the needed functionality, so there is a switching between Quartz and QuickDraw for some things, Lyons: pg. 1). However, Lyons teaches exactly the same.

Morgenstern, Portuesi, Erickson, Apple2, Lipton and Lyons do not explicitly teach displaying at least a portion of the application content in an opaque content portion of the windows (fully opaque window with transparent sub-parts, Moki: sixth paragraph). However, Moki teaches exactly the same. Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to use different functionalities of different versions of Jaguar operating system as taught by Morgenstern, Portuesi,

Erickson, Apple2, Lipton, Lyons and Moki because combining different functionalities will result in a better and more user-friendly operating system.

28. Regarding claim 19, Moki teaches the frame portion is translucent when the window has an input focus (it should be noted that the title bar attached to the window has a level of transparency associated with them, so even a fully opaque window has transparent sub-parts such as the title bar, Moki: paragraph five and six).

29. Regarding claim 21, the statements presented above with respect to claim 1 are incorporated herein.

Although Jaguar teaches the limitations as stated above in claim 1, Jaguar does not explicitly teach a computer readable medium storing computer executable instructions that cause a computer to perform a method of rendering a desktop window in a graphical user interface of an operating system. However, the examiner takes an official notice of the fact that it was known to one of ordinary skill in art at the time of present invention to execute a programmable process stored on use a computer readable medium because by using a portable computer readable medium to store a process that can be executed by the computer allows to perform the execution of the process on any computer and therefore provides portability and reusability.

30. Regarding claim 39, the statements presented above with respect to claims 21 and 19 are incorporated herein.

31. Claims 7-11, 15-16, 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki,

and further in view of John Siracusa (Mac OS X 10.2 Jaguar; http://arstechnica.com/reviews/os/macosx-10.2.ars/8; September 5, 2002; pgs. 1-5; hereinafter Siracusa).

32. Regarding claims 7 and 8, Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki do not explicitly teach the CDWM modeling the window by applying a texture to a mesh (window/polygon). However, Siracusa teaches exactly the same (it should be noted that a mesh according to the specification is 2D or 3D primitive, see paragraph [0015] on pg. 15; it should be further noted that each window is treated as an OpenGL surface and the texture is mapped onto that surface, Siracusa: pg. 3 second paragraph and pg. 4 first five lines; the window server, now an OpenGL application itself, retains the resulting bitmaps as textures on polygons in an OpenGL scene and composites them into a pleasing, cohesive final image on the screen, Siracusa: pg. 3 second paragraph). Siracusa further teaches the mesh is defined by a current visual style (each window is drawn according to its position and layering; each window is represented as a bitmap that includes alpha channel and anti-aliasing information; thus the position and layering of each window will give its current alpha channel information, which defines its current visual style; Siracusa: pg. 1 last paragraph). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to combine the teachings of Siracusa into the operating system of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki because such an operating system will composite the resulting bitmaps as textures on polygons and composite them into a pleasing, cohesive final image on the screen.

33. Regarding claim 9, Siracusa teaches the mesh is provided in the application content information (each window and its associated bitmap is provided to the Quartz compositor; Siracusa: pg. 1 last two paragraphs, pg. 3 second paragraph and pg. 4 first five lines). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to combine the teachings of Siracusa into the operating system of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki because such an operating system will composite the resulting bitmaps as textures on polygons and composite them into a pleasing, cohesive final image on the screen.

34. Regarding claim 10, Siracusa teaches the texture is defined by a current visual style (each window is drawn according to its position and layering; each window is represented as a bitmap that includes alpha channel and anti-aliasing information; the bitmap that makes up the window's contents is the texture mapped on that surface; thus the position and layering of each window and it's associated bitmap texture will give its current alpha channel information, which defines its current visual style; Siracusa: pg. 1 last paragraph, pg. 3 second paragraph and pg. 4 first five lines). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to combine the teachings of Siracusa into the operating system of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki because such an operating system will composite the resulting bitmaps as textures on polygons and composite them into a pleasing, cohesive final image on the screen.

35. Regarding claim 11, Siracusa teaches the texture is provided in the application content information (each window is drawn according to its position and layering; each

window and it's associated bitmap texture is provided to the Quartz compositor to composite them into a pleasing, cohesive final image on the screen; Siracusa: pg. 1 last two paragraphs, pg. 3 second paragraph and pg. 4 first five lines). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to combine the teachings of Siracusa into the operating system of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki because such an operating system will composite the resulting bitmaps as textures on polygons and composite them into a pleasing, cohesive final image on the screen.

36. Regarding claim 15, Lyons teaches the switching is based on the current visual style (pixels displayed belong to one window or the desktop and switch immediately when moving from one window to another, Cocoa NSView's can be set to use a certain cursor image whenever the cursor is over them; Cocoa application that uses Quartz 2D does not provide all the needed functionality, so there is a switching between Quartz and QuickDraw for some things based on the current visual style of the area over which mouse is moved, Lyons: pg. 1 and pg. 2 last paragraph).

37. Regarding claim 16, Lyons teaches the switching is based on a current configuration of a computer on which the method is being performed (the front application switches the cursor in response to mousing over an area receiving an mouse moved event and telling the window server to change the cursor; Cocoa NSView's can be set to use a certain cursor image whenever the cursor is over them; Cocoa application that uses Quartz 2D does not provide all the needed functionality, so there is a switching between Quartz and QuickDraw for some things based on the

current visual style of the area over which the mouse was moved; the visual style of this area is the visual configuration of the computer on which the method is being performed, Lyons: pg. 1 and pg. 2 last paragraph). It should also be noted that Siracusa also teaches (Quartz compositor composites all of the visible window bitmaps with each other according to their position and layers; each window bitmap includes alpha channel and anti-aliasing information, Siracusa: pg. 1 last two paragraphs).

38. Regarding claim 27, the statements presented above with respect to claims 21 and 7 are incorporated herein.

39. Regarding claim 28, the statements presented above with respect to claims 21 and 8 are incorporated herein.

40. Regarding claim 29, the statements presented above with respect to claims 21 and 9 are incorporated herein.

41. Regarding claim 30, the statements presented above with respect to claims 21 and 10 are incorporated herein.

42. Regarding claim 31, the statements presented above with respect to claims 21 and 11 are incorporated herein.

43. Claim 44 is rejected under 35 U.S.C. 103(a) as being unpatentable over Erickson, in view of Lyons and further in view of Siracusa.

44. Regarding claim 44, Erickson teach the instance of the legacy application program (Carbon/Cocoa application) providing legacy window information from an instance of a legacy application program to a legacy desktop window manager (DWM)

(QuickDraw) executing on the computer (picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D, Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; it should be noted that examiner takes an official notice of the fact that QuickDraw is a legacy API from classic Mac OS). Erickson further teaches stripping out client content from the legacy window information (QuickDraw receives the picture content from the picture window of the Carbon application, Erickson: pg. 1 last two lines and pg. 2 first three lines), and converting the client content to raster image of the client content (QuickDraw generates the graphical representation of the application data using it's drawing methods; picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D, Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; it should be also noted that picture window is a Carbon application which uses QuickDraw's drawing methods and text window is a Cocoa application which uses Quartz2D, Erickson: pg. 1 last two lines and pg. 2 first four paragraphs; It should be noted that Cocoa application that uses Quartz 2D does not provide all the needed functionality, so there is a switching between Quartz and QuickDraw for some things, Lyons: pg. 1).

Although Erickson teaches the limitations as stated, Erickson does not explicitly teach a compositing desktop window manager (CDWM) (Quartz 2D and Quartz compositor), executing on the computer, drawing a window to a buffer memory (Siracusa: pg. 1 last paragraph, pg. 2 first two paragraphs and figure on pg. 3), wherein the CDWM renders the window by applying a texture to a mesh (window/polygon) (it

should be noted that a mesh according to the specification is 2D or 3D primitive, see paragraph [0015] on pg. 15; it should be further noted that each window is treated as an OpenGL surface and the texture is mapped onto that surface, Siracusa: pg. 3 second paragraph and pg. 4 first five lines). However Siracusa teaches exactly the same (the window server, now an OpenGL application itself, retains the resulting bitmaps as textures on polygons in an OpenGL scene and composites them into a pleasing, cohesive final image on the screen, Siracusa: pg. 3 second paragraph). Siracusa further teaches wherein the texture comprises the raster image (bitmap) of the client content and the default non-client information (bitmap includes translucency and anti-aliasing information, Siracusa: pg. 1 last paragraph, pg. 2 first two paragraphs, pg. 3 second paragraph, pg. 4 first paragraph; all of the bitmapped data produced by QuickDraw is passed on to the Quartz Compositor for eventual display on the screen, Siracusa: pg. 1 seventh paragraph, figure on pg. 3 and 4). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to combine the teachings of Siracusa into the operating system of Erickson and Lyons because such an operating system will composite the resulting bitmaps as textures on polygons and composite them into a pleasing, cohesive final image on the screen. Further, it would have been obvious to one of ordinary skill in the art to use different functionalities of different versions of Jaguar operating system as taught by Erickson, Lyons and Siracusa because combining different functionalities will result in a better and more user-friendly operating system.

45. Claims 18 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki, and further in view of Solazzi (US 2003/0107570).

46. Regarding claim 18, although Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki teach the limitations as stated, they do not explicitly teach the frame comprises reflective content based on other content in the graphical user interface separate from the window. However, Solazzi teaches a 3D image (window frame) can reflect the surroundings (other content in the graphical user interface separate from the window) ([0008]). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have 3D images with reflective characteristics as taught by Solazzi and apply into the operating system of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki because such reflective characteristics illustrate the ability of the object to reflect light ([0008]).

47. Regarding claim 38, the statements presented above with respect to claims 21 and 18 are incorporated herein.

48. Claims 17 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki, in view of Solazzi, and further in view of Whitman (Technology Terminology, Mike Whitman, May 13, 2001, http://web.archive.org/web/20010513215002/http://bigelowmiddleschool.com/programs/Teched/techterms.html).

49. Regarding claim 17, although Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki teach the limitations as stated, they do not explicitly teach the frame comprises spectral highlights based on a virtual light source. However, Solazzi teaches a 3D image (window frame) can reflect the surroundings ([0008]; it should be noted that Whitman defines spectral highlight as a bright reflection from a light source containing little or no detail; Solazzi teaches a 3D image can display reflective characteristics, and therefore the reflective characterisitics of the 3D image of Solazzi also includes spectral highlights which is a bright reflection form a light source). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have 3D images with reflective characteristics as taught by Solazzi and apply into the operating system of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki because such reflective characteristics illustrate the ability of the object to reflect light ([0008]).

50. Regarding claim 37, the statements presented above with respect to claims 21 and 17 are incorporated herein.

51. Claim 41 is rejected under 35 U.S.C. 103(a) as being unpatentable over Morgenstern, in view of Moki, in view of Solazzi, in view of Whitman, and further in view of Fowler et al. (US 2002/0180741; hereinafter Fowler).

52. Regarding claim 41, Morgenstern teaches a computer implemented method for rendering a desktop window in a graphical user interface of an operating system shell, comprising: receiving application content to display in a window (Quartz compositor takes information from the rendering component and writes it on the screen,

Morgenstern: pg. 1 last paragraph). Morgenstern further teaches window having a frame portion (translucent title bars of inactive windows, Morgenstern: pg. 2 third paragraph). Morgenstern further teaches the compositing desktop window manager is configured to provide transparency and shadows (Quartz's window server makes it easy to see the outlines and shadings of buttons and other window elements through the translucent title bars of inactive windows, Morgenstern: pg. 2 paragraph 3, figure 1).

Although Morgenstern teaches the limitations as stated, Morgenstern does not explicitly teach displaying at least a portion of the application content in a content portion of the window having a frame portion (fully opaque window with transparent sub-parts, Moki: sixth paragraph; transparent sub-parts such as title bars, shadows, etc., Moki: sixth paragraph; Moki further teaches fully opaque windows has transparent sub-parts such as title bar, the shadows, etc, Moki: paragraphs 3, 4, 6 and 9). . Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to use different functionalities of different versions of Jaguar operating system as taught by Morgenstern and Moki because combining different functionalities will result in a better and more user-friendly operating system.

Although Morgenstern and Moki teach the limitations as stated, they do not explicitly teach rendering spectral highlights on the frame portion based on a virtual light source by the compositing desktop window manager configured to provide light effects. However, Solazzi teaches a 3D image (window frame) can display reflective and refractive characteristics ([0008]; it should be noted that the unit performing 3D modeling corresponds to compositing desktop window manager; it should be noted that

Whitman defines spectral highlight as a bright reflection from a light source containing little or no detail; Solazzi teaches a 3D image can display reflective characteristics, and therefore the reflective characterisitics of the 3D image of Solazzi also includes spectral highlights which is a bright reflection form a light source; spectral highlights correspond to lighting effects; therefore, compositing desktop window manager provides lighting effects). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have 3D images with refractive characteristics as taught by Solazzi and apply into the operating system of Morgenstern and Moki because refractive properties added to the 3D image makes the image appear more realistic and reflective characteristics illustrate the ability of the object to reflect light ([0008], [0014]).

Although Morgenstern, Moki, Solazzi and Whitman teach the limitations as stated above, they do not explicitly the compositing desktop window manager is configured to provide bump mapping and environment mapping. However, Fowler teaches OpenGL supports environment-mapped bump mapping (environment-mapped bump mapping corresponds to bump mapping and environment mapping, [0038-0039]). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to provide bump mapping and environment mapping as taught by Fowler to the method Morgenstern, Moki, Solazzi and Whitman because such an approach will achieve a rendered surface whose appearance changes more realistically with lighting conditions ([0038]).

53. Claim 43 is rejected under 35 U.S.C. 103(a) as being unpatentable over Morgenstern, Portuesi and Moki, in view of Solazzi, in view of Whitman, in view of Fowler, and further in view of Ben-Shachar et al. (US 2003/0189599; hereinafter Ben-Shachar).

54. Regarding claim 43, Morgenstern teaches a computer implemented method for rendering a desktop window in a graphical user interface of an operating system shell, comprising: receiving application content, at a compositing desktop window manager (CDWM) (receiving the application content is taught by Morgenstern: pg. 1 last paragraph, pg. 3 third and fourth paragraphs). It should be noted that Morgenstern further teaches QuickDraw handles text, vector graphics and bitmapped images, and then send them to the screen and output devices (Quartz compositor takes information from the rendering component and writes it on the screen, Morgenstern: pg. 1 fourth paragraph and last paragraph). Morgenstern further teaches the windows having translucent frame portions (translucent title bars of inactive windows, Morgenstern: pg. 2 third paragraph). Morgenstern further teaches the compositing desktop window manager is configured to provide transparency and shadows (Quartz's window server makes it easy to see the outlines and shadings of buttons and other window elements through the translucent title bars of inactive windows, Morgenstern: pg. 2 paragraph 3, figure 1).

However, Morgenstern does not explicitly teach receiving application content in a bottom-to-top order (video from VL is passed on to OpenGL by converting it into bottom-to-top orientation from top-to-bottom orientation, Portuesi: pg. 2-3), to display the

application content received in a bottom-to-top order in windows (openGL renders in bottom-to-top orientation, Portuesi: pg. 2; Quartz 2D renders drawing primitives, PDF documents, text and images using bottom-to-top operation, Lindberg: pg. 1). However, Portuesi teaches exactly the same.

Although Morgenstern and Portuesi teach the limitations as stated, they do not explicitly teach displaying at least a portion of the application content in a content portion of the window having a frame portion (fully opaque window with transparent sub-parts, Moki: sixth paragraph; transparent sub-parts such as title bars, shadows, etc., Moki: sixth paragraph; Moki further teaches fully opaque windows has transparent sub-parts such as title bar, the shadows, etc, Moki: paragraphs 3, 4, 6 and 9). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to use different functionalities of different versions of Jaguar operating system as taught by Morgenstern, Portuesi and Moki because combining different functionalities will result in a better and more user-friendly operating system.

Although Morgenstern, Portuesi and Moki teaches the limitations as stated above, they do not explicitly teach rendering refractive content on the frame portion based on the other discrete content behind the window in the graphical user interface by the compositing desktop window manager, which is configured to provide light effects. However, Solazzi teaches a 3D image (window frame) can display reflective and refractive characteristics ([0008]; it should be noted that the unit performing 3D modeling corresponds to compositing desktop window manager; it should be noted that Whitman defines spectral highlight as a bright reflection from a light source containing

little or no detail; Solazzi teaches a 3D image can display reflective characteristics, and therefore the reflective characterisitics of the 3D image of Solazzi also includes spectral highlights which is a bright reflection form a light source; spectral highlights correspond to lighting effects; therefore, compositing desktop window manager provides lighting effects). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have 3D images with refractive characteristics as taught by Solazzi and apply into the operating system of Morgenstern, Portuesi and Moki because refractive properties added to the 3D image makes the image appear more realistic and reflective characteristics illustrate the ability of the object to reflect light ([0008], [0014]).

Although Morgenstern, Portuesi, Moki, Solazzi and Whitman teach the limitations as stated above, they do not explicitly the compositing desktop window manager is configured to provide bump mapping and environment mapping. However, Fowler teaches OpenGL supports environment-mapped bump mapping (environment-mapped bump mapping corresponds to bump mapping and environment mapping, [0038-0039]). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to provide bump mapping and environment mapping as taught by Fowler to the method Morgenstern, Portuesi, Moki, Solazzi and Whitman because such an approach will achieve a rendered surface whose appearance changes more realistically with lighting conditions ([0038]).

Although Morgenstern, Portuesi, Moki, Solazzi, Whitman and Fowler teach the limitations as stated above, they do not explicitly teach receiving the application content in reverse z-order. However, Ben-Shachar teaches exactly the same ([0061]; it should

be noted that DT_WINDLST packet containing a list corresponds to application content; it should be noted that the process proceeds through all windows in the windows list in reverse z-order). Therefore, it would have been obvious to one of ordinary skill in the art at the time of present invention to process through the windows list in reverse z-order as taught by Ben-Shachar and apply it to the method of Morgenstern, Portuesi, Moki, Solazzi, Whitman and Fowler because treating the window information is such a manner will create the viewer display of shared window ([0061]).

55. Claim 43 is rejected under 35 U.S.C. 103(a) as being unpatentable over Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki, in view of Solazzi, in view of Whitman, in view of Fowler, and further in view of Ben-Shachar.

56. Regarding claim 42, the statements presented above with respect to claims 1, 18 and 43 are incorporated herein.

57. Claims 2 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki, and further in view of Donham et al (US 6980209; hereinafter Donham).

58. Regarding claim 2, Morgenstern teaches distorting content on top of which the frame portion is rendered (contents underneath the translucent title bars of inactive windows can be seen, Morgenstern: pg. 2 paragraph three and fig. 1 and it's description; thus when a title bar is translucent, blending needs to be performed on the title bar and contents of another window underneath the translucent title bar). Further,

Moki teaches a window can have shadow and a title bar (the shadow and title bar has a level of transparency associated with them, Moki: paragraph six and seven).

Although Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki teach the limitations as stated, they do not explicitly teach a pixel shader is needed to perform the blending. However, Donham teaches a pixel shader that blends the texels with the color values of the pixels to be textured (col. 5 lines 25-35). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to use a pixel shader to perform blending as taught by Donham into the method of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons and Moki because a pixel shader combines pixel data and texture data to produce the combined pixel data (col. 5 lines 25-31).

59. Regarding claim 22, the statements presented above with respect to claims 21 and 2 are incorporated herein.

60. Claims 20, 40 and 45-48 rejected under 35 U.S.C. 103(a) as being unpatentable over Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons, Moki and Siracusa, in view of Farrah (US 2004/0030997), and further in view of Meagher (US 4694404).

61. Regarding claims 20, 40 and 45-47, Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons, Moki and Siracusa teach the limitations as stated, except that they do not explicitly teach receiving user input to resize the window and dividing the mesh into the three regions per mesh dimension. However, Farrah teaches to resize the window by dividing it into several equally sized and not equally sized regions based on the user input (fig. 21a-c, fig. 22a-c, [0225-0227], [0231-0235]; it should be noted that selecting

the number of rows as "3" will divide the window in nine equal-sized regions; it should be noted that dividing window is functionally equivalent to resizing the window; it should be noted that dividing window is functionally equivalent to resizing the window; it should be noted that the window is being divided along x and y dimensions). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to divide a window mesh into several regions as taught by Farrah and apply it into the method of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons, Moki and Siracusa because such regions are commonly used in computer programs which are used to generate artworks, drawings and flow charts ([0006]).

Although the combination of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons, Moki, Siracusa and Farrah teach the limitations as stated above, they do not explicitly teach for each region, maintaining offsets of mesh vertices in any dimension by which the region is bounded by a bounding box of the window, and scaling mesh vertices in any dimension by which the region is not bounded by the bounding box of the window. However, Meagher shows offsets (maintaining offsets of mesh vertices) from each line correspond to vertices of each of the four windows, and the critical vertices for a window overlay selected from 3 x 3 array may be calculated by adding offsets (scaling the vertices) as a function of the value in x and y directions (fig. 6a-f, fig. 23a-e, col. 8 lines 31-51, col. 58 lines 64-67, col. 59 lines 1-15, col. 60 lines 3-21 and lines 56-67, col. 61 lines 1-17). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to scale the vertices as taught by Meagher and apply it into the method of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons,

Moki, Siracusa and Farrah because such a method scales the three-dimensional universe relative to the three-dimensional coordinate system using the independent scaling factors for each of the x, y and z directions input by floating point multiplication on the microcomputer (col. 60 lines 56-60).

62. Regarding claim 48, although the combination of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons, Moki, Siracusa and Farrah teach the limitations as stated above, they do not explicitly teach regions bounded by the bounding box are as small as necessary to encompass material that should not be scaled. However, Meagher teaches the bounding box of the node projection is the same size or smaller (in each dimension) as a window at that level depending on the size of the node projection as determined by the user specified scale factor (the size of the bounding box depends on the scale factor determined by the use, col. 44 lines 57-67, col. 45 lines 1-5). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the bounding box size is determined by the scale factor as taught by Meagher and apply it into the method of Morgenstern, Portuesi, Erickson, Apple2, Lipton, Lyons, Moki, Siracusa and Farrah because it helps to determine if the node projection intersects any non-full window in the current window overlay (col. 45 lines 6-10).

Conclusion

63. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JWALANT AMIN whose telephone number is (571)272-2455. The examiner can normally be reached on 10:30 a.m. - 7:00 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kee Tung can be reached on 571-272-7794. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Kee M Tung/
Supervisory Patent Examiner, Art Unit 2628

/J. A./
Examiner, Art Unit 2628